Array.prototype.sort()

The sort() method sorts the elements of an array in place and returns the sorted array. The default sort order is ascending, built upon converting the elements into strings, then comparing their sequences of UTF-16 code units values.

Syntax

```
// Functionless
sort()
// Arrow function
sort((firstEl, secondEl) => { ... } )
// Compare function
sort(compareFn)
// Inline compare function
sort(function compareFn(firstEl, secondEl) { ... })
```

The `sort` method can be conveniently used with function expressions:

```
var numbers = [4, 2, 5, 1, 3];
numbers.sort(function(a, b) {
  return a - b;
});
console.log(numbers);

// [1, 2, 3, 4, 5]
```

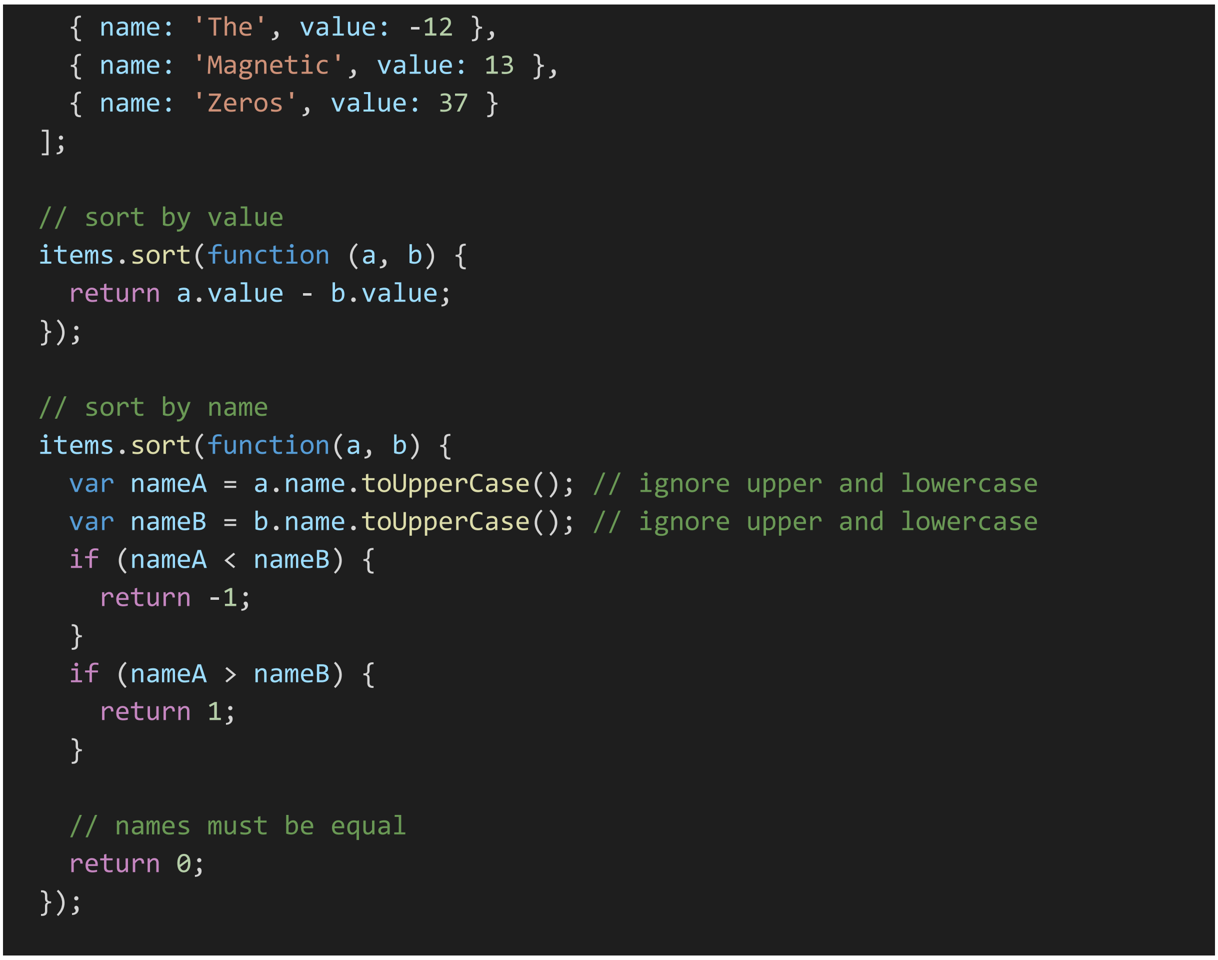
provides arrow function expressions with even shorter syntax.

```
let numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b);
console.log(numbers);

// [1, 2, 3, 4, 5]
```

Arrays of objects can be sorted by comparing the value of one of their properties.

```
var items = [
    { name: 'Edward', value: 21 },
    { name: 'Sharpe', value: 37 },
    { name: 'And', value: 45 },
```



```
  { name: 'The', value: -12 },
  { name: 'Magnetic', value: 13 },
  { name: 'Zeros', value: 37 }
];

// sort by value
items.sort(function (a, b) {
  return a.value - b.value;
});

// sort by name
items.sort(function(a, b) {
  var nameA = a.name.toUpperCase(); // ignore upper and lowercase
  var nameB = b.name.toUpperCase(); // ignore upper and lowercase
  if (nameA < nameB) {
    return -1;
  }
  if (nameA > nameB) {
    return 1;
  }

  // names must be equal
  return 0;
});
```

Array.prototype.findIndex()

The **findIndex()** method returns the **index** of the first element in the array **that satisfies the provided testing function**. Otherwise, it returns -1, indicating that no element passed the test.

Syntax

```
// Arrow function
findIndex((element) => { ... } )
findIndex((element, index) => { ... } )
findIndex((element, index, array) => { ... } )

// Callback function
findIndex(callbackFn)
findIndex(callbackFn, thisArg)

// Inline callback function
findIndex(function callbackFn(element) { ... })
findIndex(function callbackFn(element, index) { ... })
findIndex(function callbackFn(element, index, array){ ... })
findIndex(function callbackFn(element, index, array) { ... }, thisArg)
```

Find the index of a prime number in an array

```
function isPrime(num) {
    for (let i = 2; num > i; i++) {
      if (num % i == 0) {
        return false;
      }
    }
    return num > 1;
  }

  console.log([4, 6, 8, 9, 12].findIndex(isPrime)); // -1, not found
  console.log([4, 6, 7, 9, 12].findIndex(isPrime)); // 2 (array[2] is 7)
```

Find index using arrow function

```
const fruits = ["apple", "banana", "cantaloupe", "blueberries", "grapefruit"];

const index = fruits.findIndex(fruit => fruit === "blueberries");

console.log(index); // 3
console.log(fruits[index]); // blueberries
```